

Zertifikatsklausur 23.11.

- Klausurtyp: Open-Book-Exam
- Bearbeitungszeit: 45 Minuten (+ 15 Minuten für Download & Upload)
- Erreichbare Punkte: 45
- Bestehensgrenze: mit 25 Punkten auf jeden Fall bestanden

Abgabe bis spätestens 18:30 Uhr per E-Mail an kurse@stads.de.

Der Code muss lauffähig sein unter: Python $\geq 3.8.5$ mit numpy $\geq 1.19.2$, pandas $\geq 1.1.3$, matplotlib $\geq 3.3.2$, seaborn $\geq 0.11.0$ bzw. plotly $\geq 4.12.0$. Falls weitere Pakete oder andere Versionen verwendet werden, muss die jeweilige Version angegeben werden.

Wir schreiben in dieser Klausur mit x^y die y-ste Potenz von x. Zum Beispiel schreiben wir 3^4 für 3 hoch 4 also für $3*3*3*3=81$.

Aufgabe 1: Grundlagen (19 Punkte)

- Erstellen Sie eine Python-Datei mit dem Namen `<Nachname>_<Vorname>_exam.py` (z.B. `mustermann_max_exam.py`) und bearbeiten Sie in dieser Datei die folgende Aufgabe.

Aufgabe 1a: Get Started (2 Punkte)

- Definieren Sie die beiden Variablen `x` und `y` als `11` bzw. `4` (1 Punkt)
- Geben Sie das Quadrat des Mittelwertes, dh. $((x+y)/2)^2$ an. (Tipp: `**`, `print`) (1 Punkt)

Aufgabe 1b: Einfache Funktion (6 Punkte)

- Definieren Sie eine Funktion `mittel_quadriert`, die die zwei Variablen `x` und `y` als Input hat. (1 Punkt)
- Die Funktion soll prüfen, ob die Differenz von `x` und `y` gleich 0 ist,
 - falls dies der Fall ist: Printe `x` und `y` sind gleich. und gebe `1` zurück. (2 Punkte)
 - andernfalls, gebe das Ergebnis von $((x+y)/2)^2$ zurück. (2 Punkte)
- Werten Sie die Funktion für die Inputkombination $(x=-4, y=4)$ und $(x=-1, y=7)$ aus. (1 Punkt)

Aufgabe 1c: Datentypen (7 Punkte)

- Erstellen Sie eine Variable `infiziert` mit dem Wert 1. Konvertieren Sie `infiziert` explizit zu einem Boolean (True/False) und speichern Sie das Ergebnis als `case` (1 Punkt)
- Lassen Sie sich die Klasse des Objektes `case` ausgeben. (1 Punkt)
- Erstellen Sie eine Liste mit dem Namen `elemente` und den folgenden Einträgen: `Luft`, `Wasser`, `Feuer` (1 Punkt)
- Fügen Sie zu der Liste `elemente` den Eintrag `Erde` hinzu und speichern Sie das Ergebnis als `elemente_neu` (1 Punkt)

- Erstellen Sie ein **dictionary** mit dem Namen **weihnachtsgeschenke** und dem folgendem Mapping. (2 Punkte)
 - "Buecher" -> 2
 - "O-Saft" -> 1.5
 - "PS5" -> **False**
 - "Kekse" -> ["Anzahl", 5]
- Lassen Sie sich das Element mit dem Key **Kekse** ausgeben. (1 Punkt)

Aufgabe 1d: Schleifen (4 Punkte)

- Erstellen Sie mit einer for-Schleife die folgende Ausgabe. (4 Punkte)

```
2
24
248
24816
2481632
248163264
248163264128
```

Tipp:

```
> print("".join(["1","2"]))
12
```

Aufgabe 2: Wichtige Pakete (26 Punkte)

- Erstellen Sie ein IPython-Notebook mit dem Namen **<Nachname>_<Vorname>_exam.ipynb** (z.B. **musterman_max_exam.ipynb**) und bearbeiten Sie in dieser Datei die folgende Aufgabe.

Aufgabe 2a: Numpy (7 Punkte)

(importieren Sie numpy und pandas)

- Erstellen Sie geschickt einen Vektor **v**, der wie folgt aussieht. (1 Punkt)

[2021, 2017, 2013, 2009, 2005, ... , 1905, 1901]

- Erstellen Sie eine Matrix **ma_diag** der Dimension (12,11), die auf ihrer ersten oberen Diagonale überall eine 1 hat und sonst nur Nullen besitzt. (1 Punkt)

Hier als Beispiel für eine 4x3 Matrix:

```
[[0. 1. 0.]
 [0. 0. 1.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

- Initialisieren Sie einen Zufallszahlengenerator. (1 Punkt)
- Verwenden Sie den Zufallszahlengenerator, um einen Vektor `z` mit 12 unabhängigen Standardnormalverteilten Zufallsvariablen zu simulieren. (1 Punkt)
- Ersetzen Sie in der Matrix `ma_diag` die erste Spalte durch den erzeugten Vektor `z`. (1 Punkt)
- Wenden Sie die Exponentialfunktion elementweise auf `ma_diag` an und transformieren Sie das Ergebnis zu einem array `ma_flat` der Dimension (1,132). (1 Punkt)
- Printen Sie die Varianz von `ma_flat`. (1 Punkt)

Aufgabe 2b: Pandas Basics (9 Punkte)

Für diese Aufgabe benötigen Sie den Datensatz `diamonds.csv`.

- Importieren Sie den Datensatz `diamonds.csv` und speichern Sie diesen unter der Variablen `df` ab (2 Punkte)
- Lassen Sie sich die ersten 10 Zeilen des Datensatzes ausgeben. (1 Punkt)
- Aus wie vielen Zeilen und Spalten besteht der Datensatz? (1 Punkt)
- Überschreiben Sie jeweils `df`:
 - Überprüfen Sie ob der Datensatz fehlende Werte enthält und löschen Sie Duplikate (d.h. Zeilen mit exakt gleichen Inhalten in jeder Spalte) (1 Punkt)
 - Löschen Sie alle Zeilen in denen mindestens eine der Variablen `x`, `y`, `z` den Wert 0 annimmt. (2 Punkte)
 - Fügen Sie zu dem Datensatz eine Spalte `volume` hinzu, welche das Produkt von `x`, `y`, `z` ist. (1 Punkt)
- Wie viele Diamanten mit einem `cut` der Qualität `Premium` sind nach den Modifikationen im Datensatz enthalten? (1 Punkt)

Aufgabe 2c: Grafiken (5 Punkte)

- Filtern Sie den Datensatz `df`, sodass dieser nur Diamanten der höchsten Klarheitsstufe `IF` enthält und speichern Sie das Ergebnis als `best`. (1 Punkte)
- Erstellen Sie einen Scatterplot der Diamanten im Datensatz `best`. Dabei soll das Volumen `volume` auf der x-Achse und der Preis `price` auf der y-Achse abgebildet werden. Sie dürfen dabei ein Paket Ihrer Wahl verwenden (z.B. Pandas, Matplotlib, Seaborn, Plotly). (2 Punkte) (falls Sie die Spalte `volume` in 2b nicht erstellen konnten, verwenden Sie hier stattdessen `x`)
- Erstellen Sie eine Grafik, welche die Verteilung der verschiedenen Qualitätsstufen (`cut`) im Datensatz `df` visualisiert. (z.B. Histogramm, Balkendiagramm, ...) (2 Punkte)

Aufgabe 2d: Pandas Advanced 5 Punkte)

- Gruppieren Sie den Datensatz `df` nach der Spalte `color` und geben Sie pro Farbe den durchschnittlichen Preis an. Welche Farbe ist im Mittel am teuersten? (2 Punkte)

- Gruppieren Sie den Datensatz `df` nach der Spalte `cut`, `color` und `clarity` und aggregieren Sie die Spalte `volume` indem Sie über diese summieren. Speichern Sie das Ergebnis als `df_agg`. (2 Punkte)
- Was ist das aggregierte Volumen von Diamanten im Datensatz `df` mit den folgenden Eigenschaften `cut: Fair`, `color: D`, `clarity: SI1`? (1 Punkt)

Abgabe

Senden Sie die `beiden` von Ihnen erstellten Dateien bis Abgabeschluss (18:30 Uhr) per E-Mail an kurse@stads.de.